Microsoft®

Windows.net™

Server Family

AWS305: IIS 6.0 Rearchitecture and Reliability

Jeff Kercher
IIS6 Program Manager
Microsoft Corporation

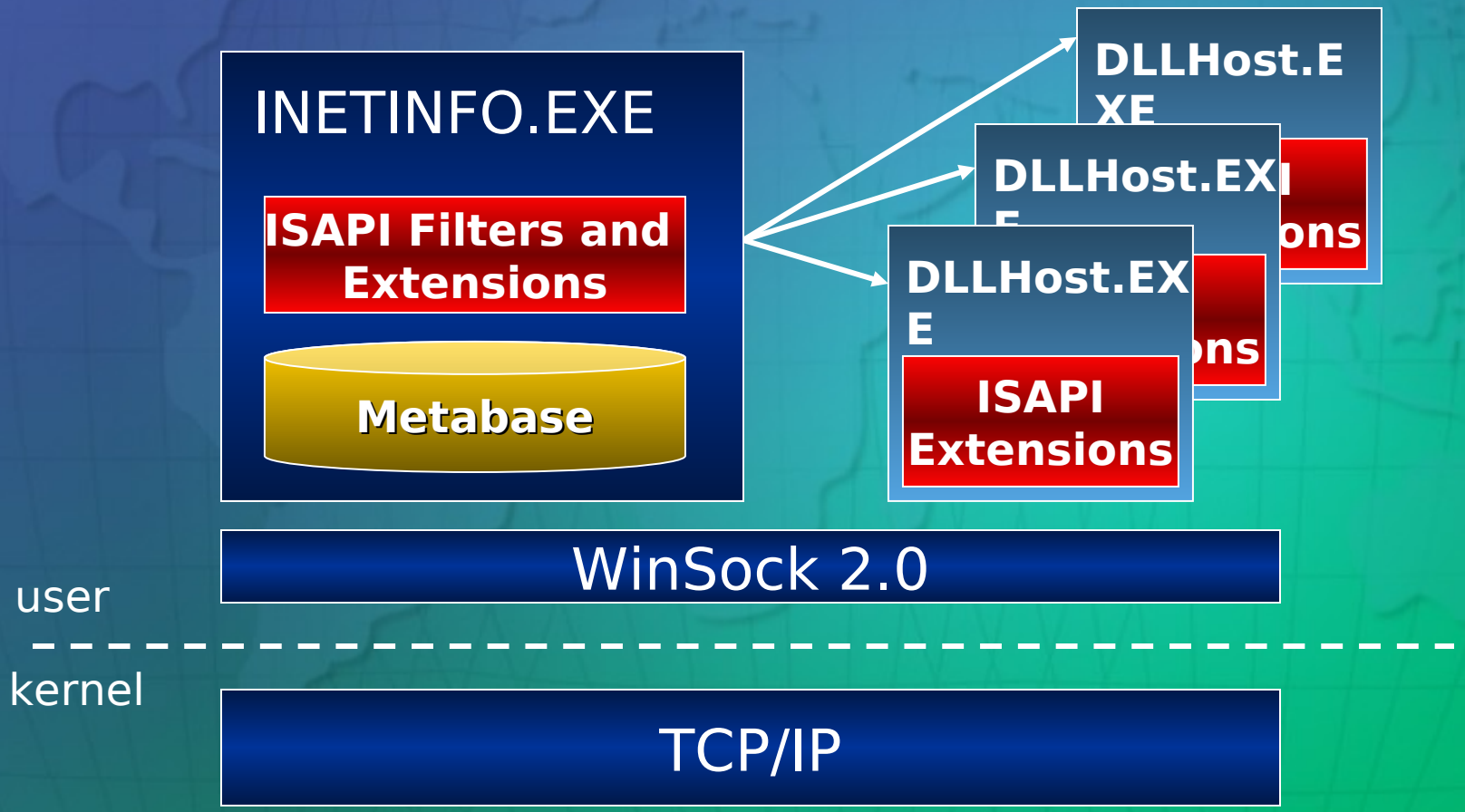
Agenda

- Rearchitecting IIS for Reliability
- Application Pooling
- Recycling
- Health Detection
- Application Tips
- Resources
- Q/A

Rearchitecting IIS
Why do it - Current Challenges

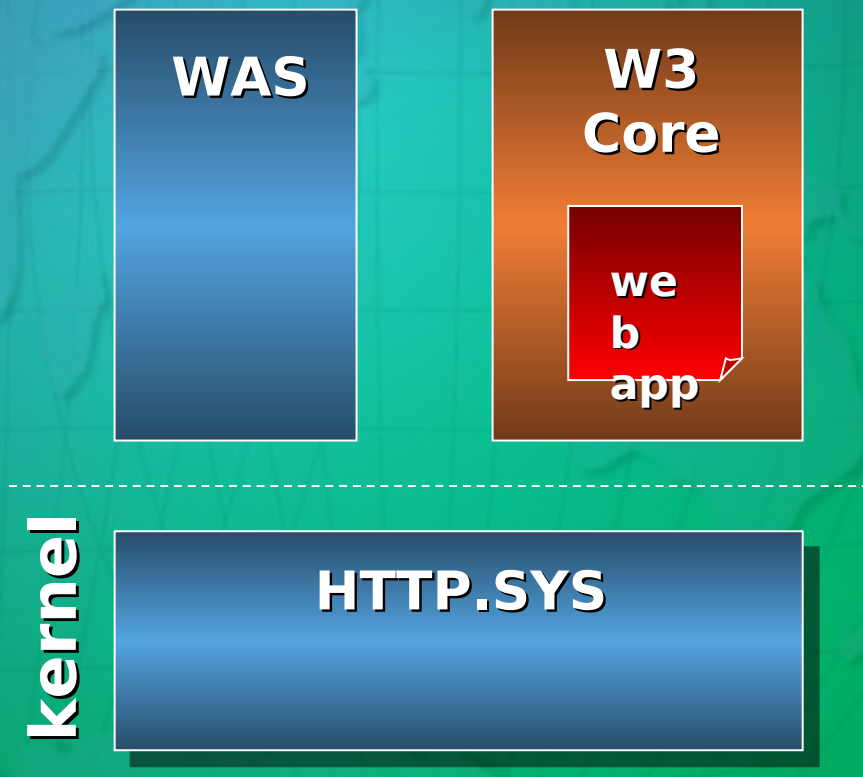
- Availability – need 99.999% uptime
  - Apps can still run in INETINFO on IIS5, which can affect web server reliability
- App isolation comes at a cost
  - Perf vs. Reliability
  - Can’t isolate everything!
- IISRESET = Heavy Hammer
  - Resets connections

Rearchitecting IIS
A review of IIS5



Rearchitecting IIS
A New Architecture for IIS6

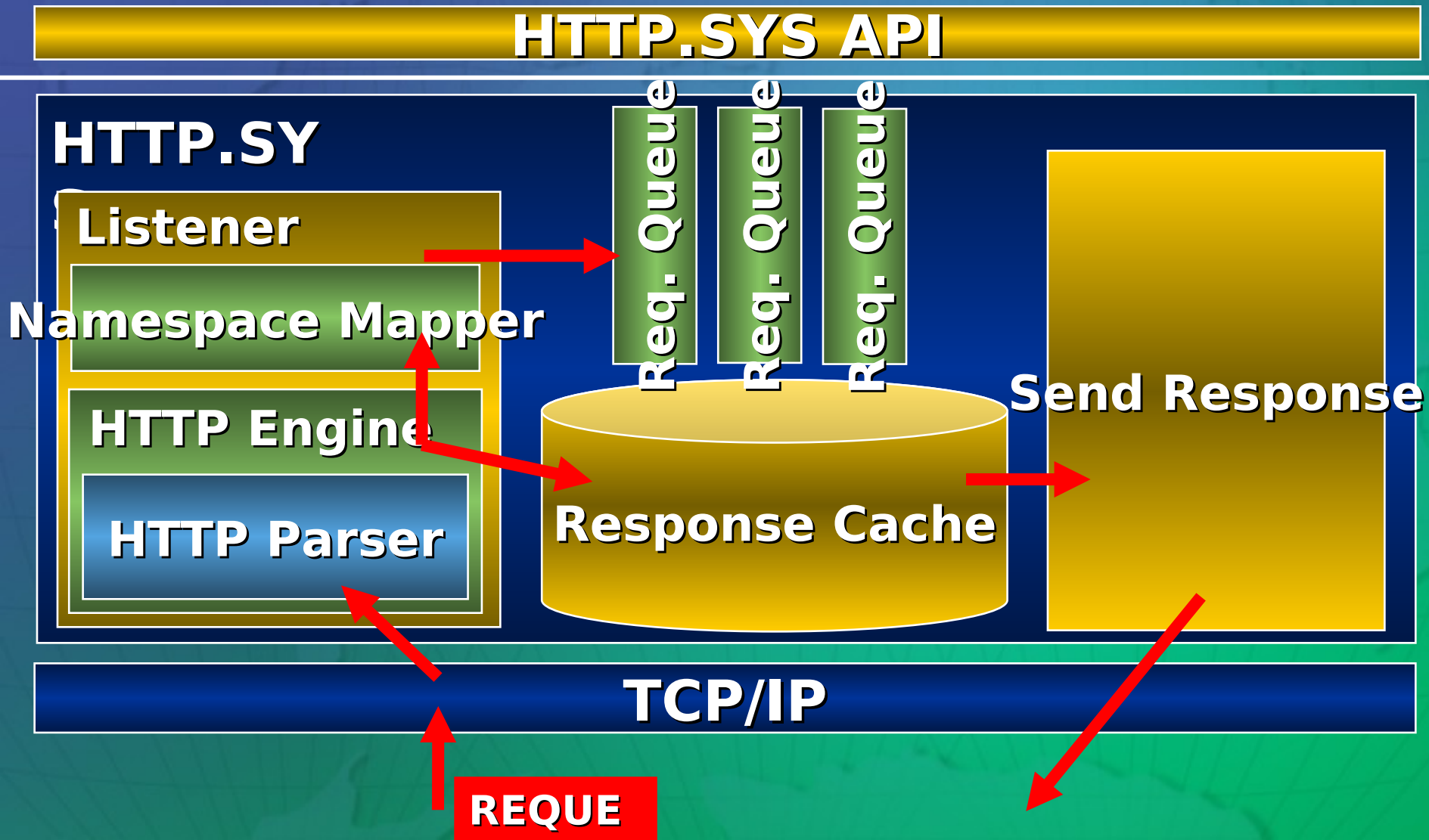
- GOAL: enable complete app isolation from other web apps and core web server
- Web service in INETINFO split out to do this:
  - HTTP.SYS: kernel mode listener and request router
  - WAS: config and process manager
  - Web Processing Core: where web apps are processed



Rearchitecting IIS
HTTP.SYS

- What is it?
  - Kernel-mode HTTP stack/listener
  - *Always* running
  - Supports listening on IPv4 and IPv6 protocols
- Reliability Features
  - Doesn’t load or run any web apps including ISAPI filters and extensions
  - Process routing based on URL
  - Request queues: kernel-mode queuing
- Performance Features
  - Kernel-mode response cache
  - Text-based and binary logging

Rearchitecting IIS
HTTP.SYS Internals



Rearchitecting IIS
Web Admin Service (WAS)

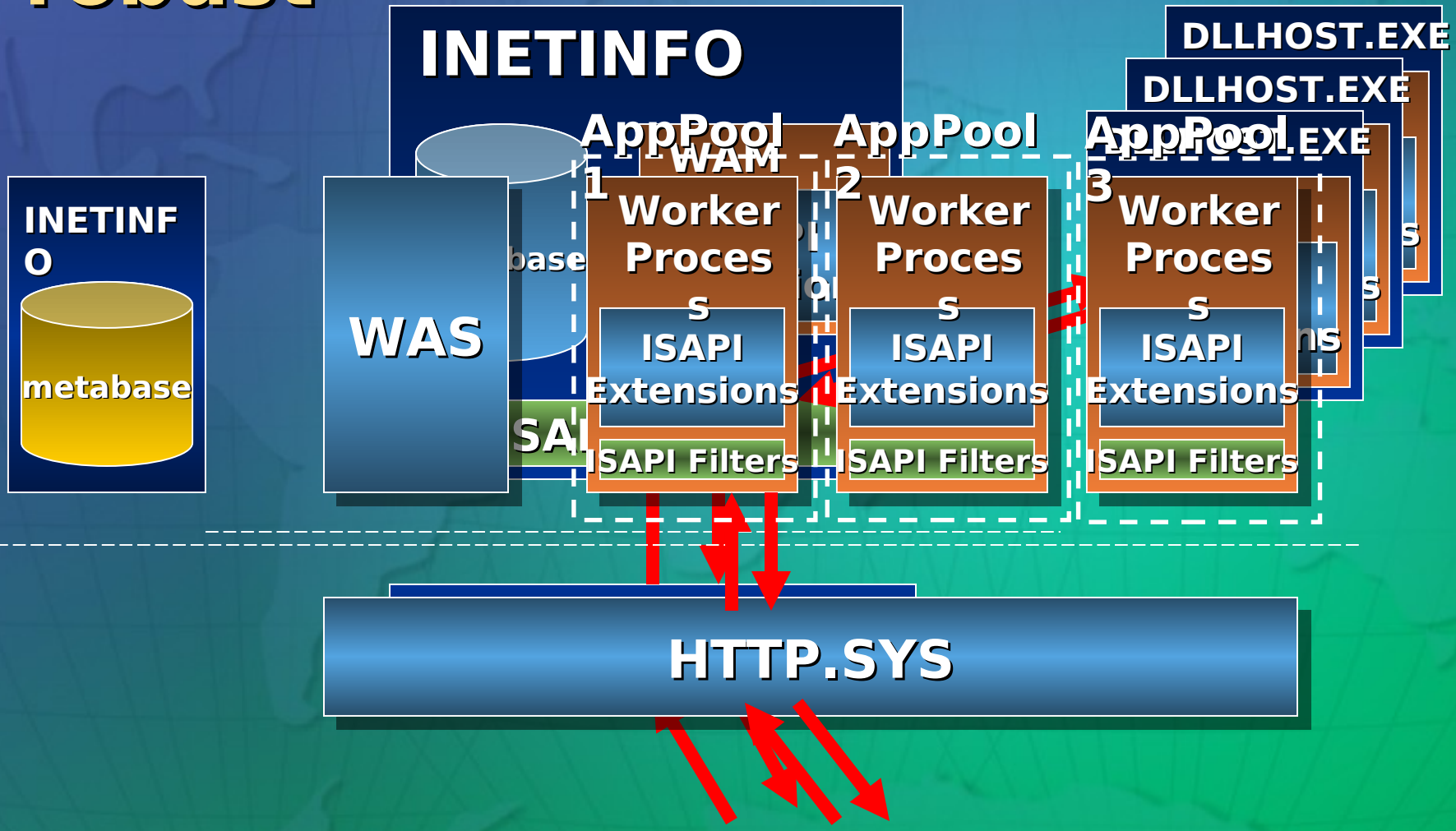
- Application Manager
  - Manages lifetime of W3 Core(s)
- Configuration Manager
  - Configures *HTTP.SYS*
- No application code
  - Ensures web server reliability
  - External monitor of web application processes
  - Ensures web application and service reliability
- Hosted in SVCHOST.exe
  - Actually part of W3SVC

Rearchitecting IIS
Web Processing Core – W3WP.exe

- What is it?
  - Main web processing core responsible for handling web requests
- Self–contained web server
  - Contains all web request processing functionality
  - Loads ISAPI's – filters and extensions
    - ASP, ASP .NET, FrontPage® Server Extensions
- Delivers complete isolation from

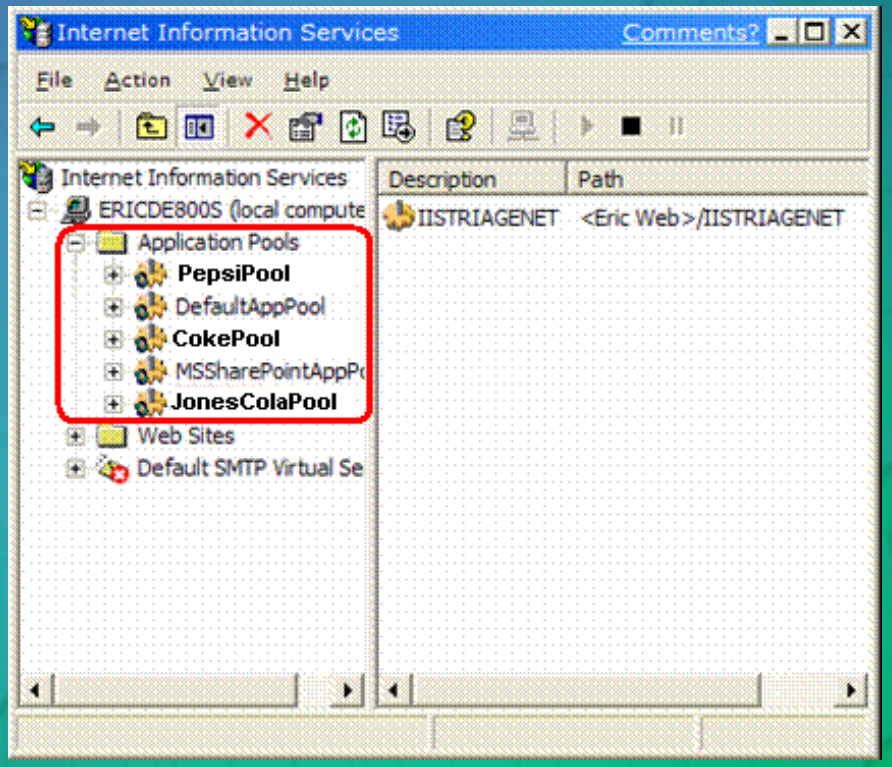
Rearchitecting IIS

IIS5 to IIS6 – Making it more robust



Application Pools
Application Isolation in Processes

- Can create 1 or more application pools
  - Each served by 1 or more processes.
  - Each worker process serves only 1 pool.
  - Reqs routed directly to pool by HTTP.sys
- Isolate apps based on:
  - Site/Customer
  - Functionality
  - Reliability



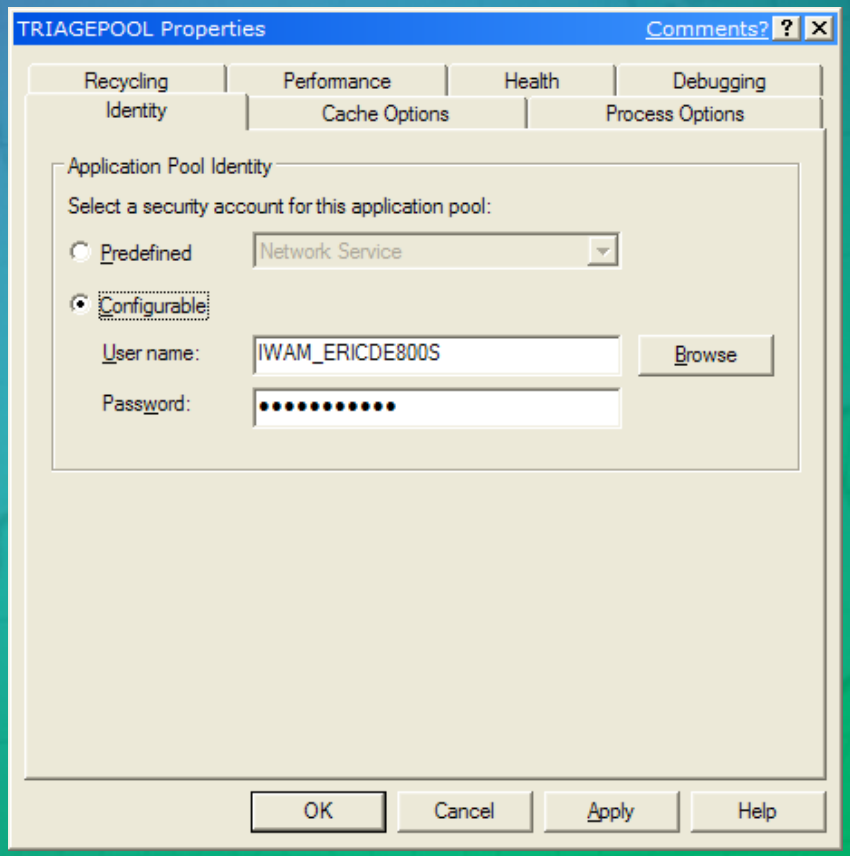
Assigning Apps to App Pools

demo

Jeff Kercher
Program Manager
IIS 6
Microsoft Corporation

Application Pooling
Configurable Worker Process ID

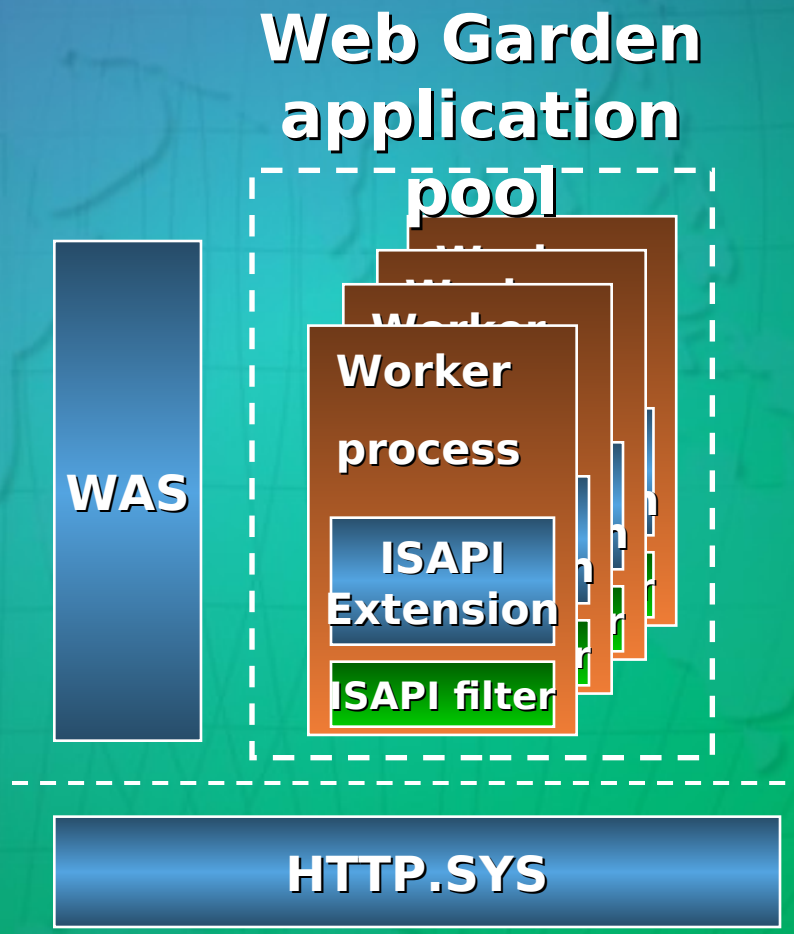
- Worker process can be started as:
  - Network Service (default)
  - Local System
  - Local Service
  - Configured ID
- IIS_WPG
- Details – Attend AWS306



Application Pooling

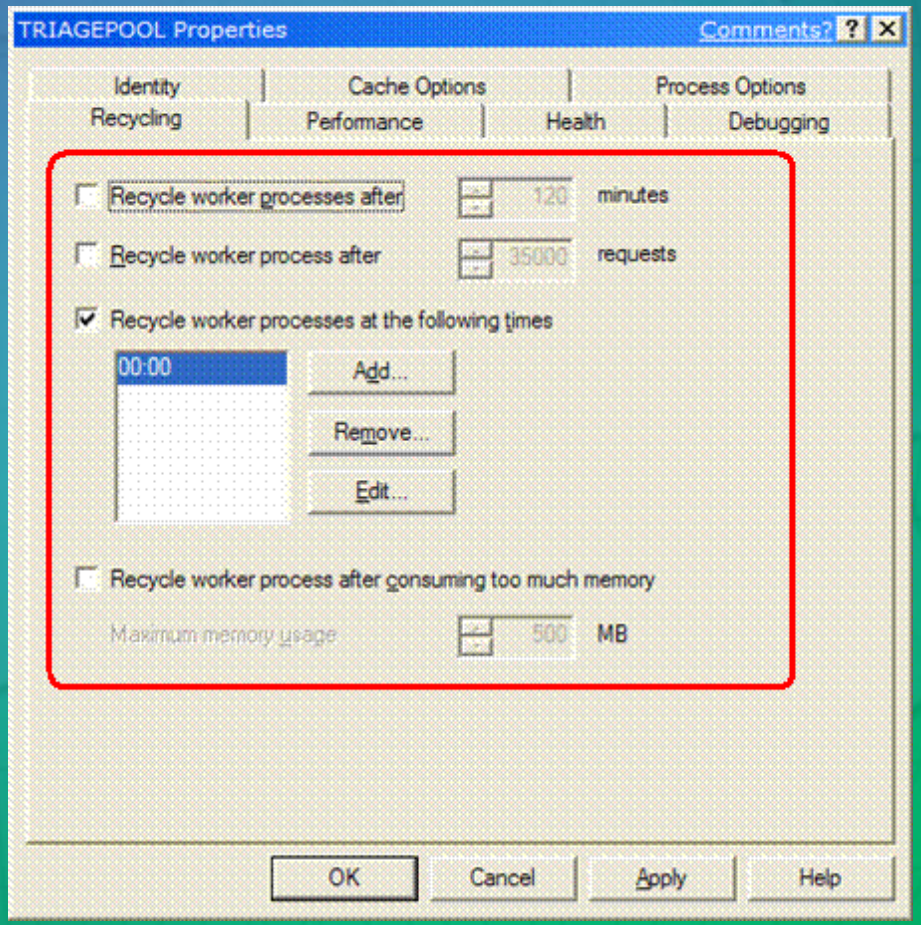
Web Gardens and Processor Affinitization

- Web Gardens
  - App Pool w/ > 1 worker process
  - Primarily reliability feature
  - Connection-based routing w/in Garden
- Processor Affinitization
  - Bind app pool process/es to 1 or more CPU's



Recycling
What is it and Why use it?

- **What is it?**
  - **Periodically restart applications based on:**
    - **Uptime**
    - **# of requests**
    - **Scheduled time**
    - **Memory consumption**
    - **On-demand**
- **Why use it?**
  - **Refresh apps to ensure availability**
  - **Prevent bad apps from taking over the system**



Recycling
Overlapping vs. Non-Overlapping

Overlapping Recycle

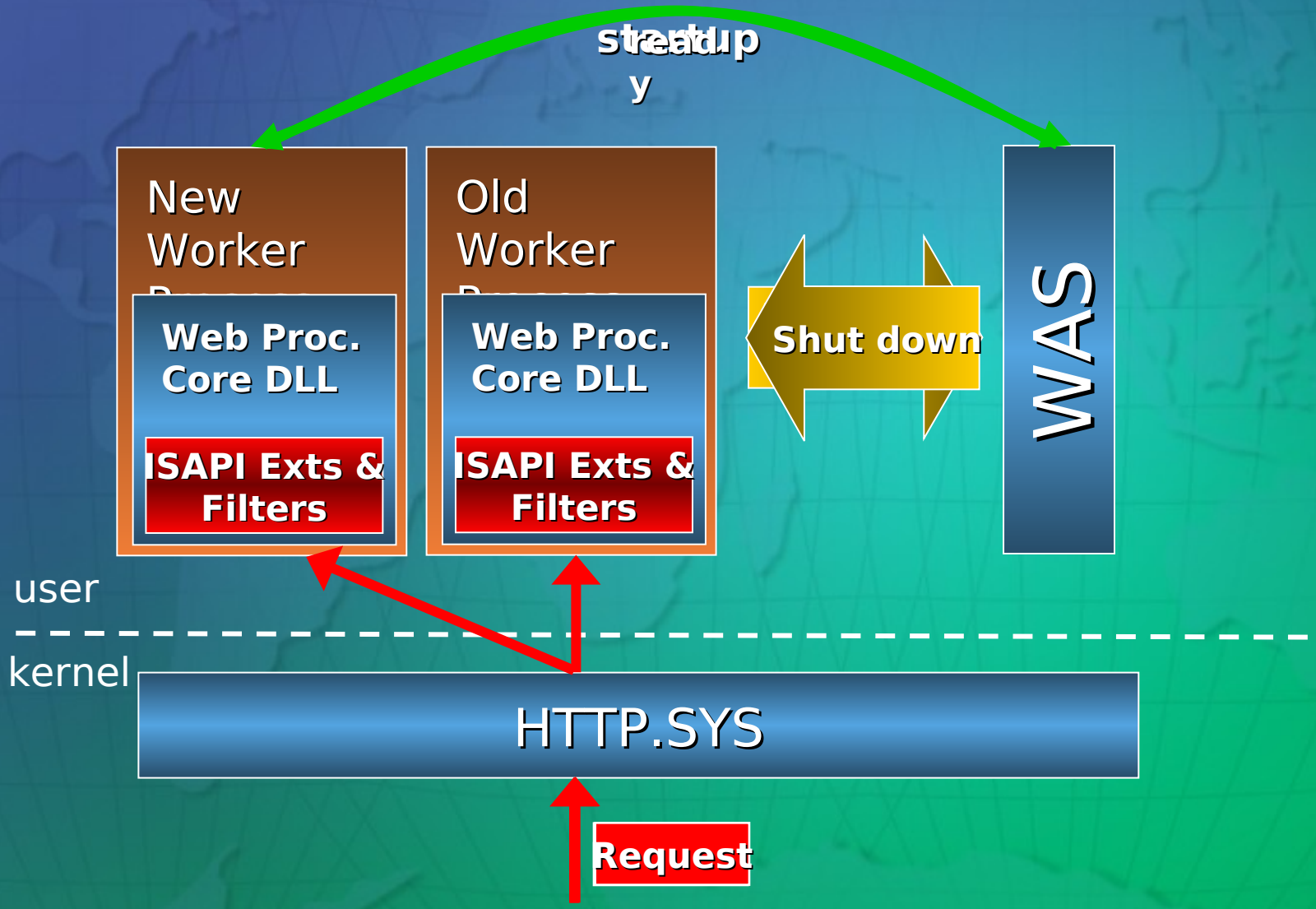
- Only terminates old process after new process starts
- Possible for apps to be instantiated multiple times

Non-Overlapping Recycle

- Terminates old process first
- *Then* starts new process to replace it
- Reduces multi-instance problems

- BOTH still queue requests in the kernel - No interruption in service!

Recycling
Overlapping Recycle



Recycling a worker process

demo

Jeff Kercher
Program Manager
IIS 6
Microsoft Corporation

Recycling
ISAPI Interaction – REPORT_UNHEALTHY

- HSE_REQ_REPORT_UNHEALTHY
  - Goal: allow an ISAPI to report to IIS that it needs to be recycled.

```
bResult = pECB-> ServerSupportFunction(
                                    pECB->ConnID,
                                    HSE_REQ_REPORT_UNHEALTHY,
                                    psz_reason_unhealthy,
                                    NULL,
                                    NULL

                                    );
```

- ASP Hang Detection
  - Used to detect when ASP threads block in components

ISAPI Report_Unhealthy

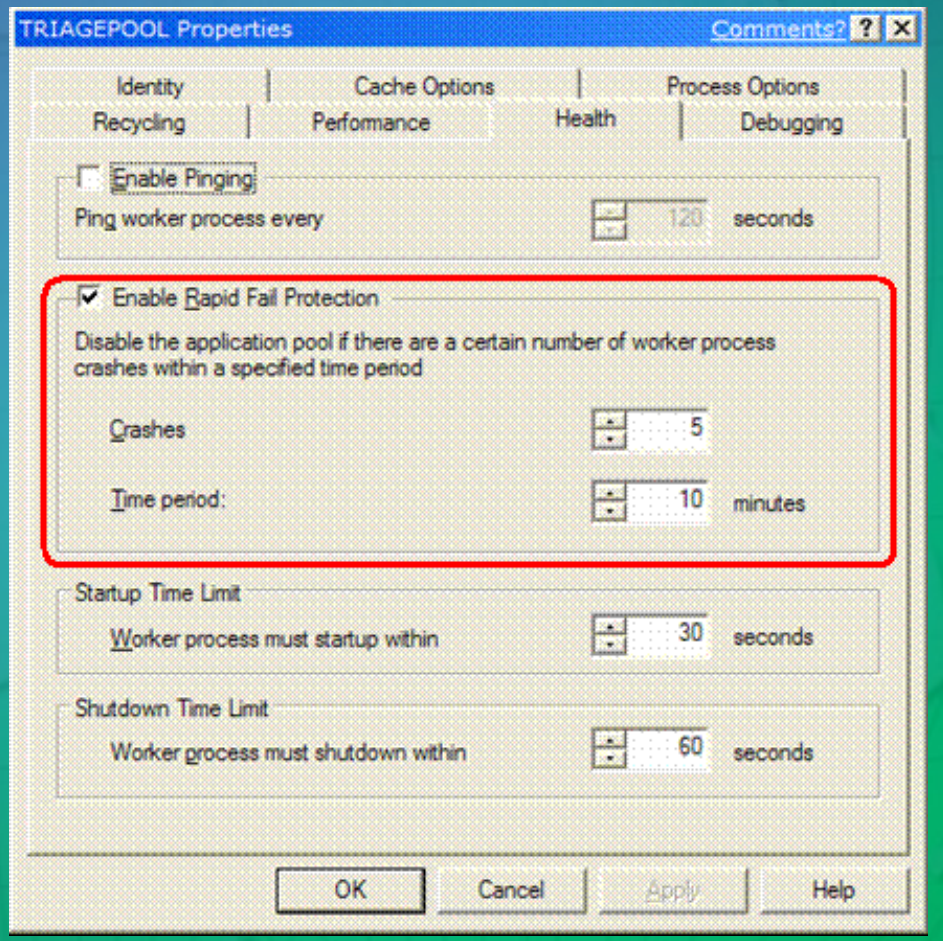
demo

Jeff Kercher
Program Manager
IIS 6
Microsoft Corporation

Health Detection

Crash Detection & Rapid Fail Protection

- WAS detects process crash/AV's
- On failure
  - Publish event to event log
  - Check "crash count"
  - If (Crash count > Max Crashes in time limit)
    - Disable app pool
  - Else start new process if demand
- Rapid Fail Protection
  - Only allow x crashes in y minutes



Rapid Fail Protection

demo

Jeff Kercher
Program Manager
IIS 6
Microsoft Corporation

Health Detection
Pinging

- What is it?
  - Designed to detect worker process thread deadlock
  - Will engage if there are no threads in worker process available to respond in time
- How does it work?
  - WAS will “ping” each worker process
  - Process has a configured time limit to respond
  - If (no response in time limit)
    - Default: kill process, publish event, and start new process

Health Detection
Configurable Debug Action

- Allows for custom action to be executed when process fails to respond – Examples:
  - Send email to admin
  - Attach debugger
  - Take process dump
- Process left running
  - Though WAS dropped its process handle

Enabling Debug Action

demo

Jeff Kercher
Program Manager
IIS 6
Microsoft Corporation

Application Tips
Considerations

- Make apps multi-instance aware
  - No single locks, etc.
  - Allows use of overlapping recycle, etc.
- Persist state externally
  - Recycled apps will lose state if persisted internal to host process
- Isolate new apps in “debug pools”
  - RFP turned on

Application Tips
ASP .NET Considerations

- ASP .NET will use the IIS6 process model when running in W3WP.exe
  - Runs completely within W3WP.exe
  - Takes advantage of complete process isolation and added health/reliability features
  - Uses IIS 6 process model settings instead of machine.config settings
- Use ASP .NET's external session state service
  - Persist state through a recycle

Summary

- IIS6 is more reliable because it:
  - Isolates apps from system components like WAS and HTTP.SYS
  - Isolates apps from one another via Application Pools
- Application Pools introduce new features to further improve reliability:
  - Periodic process recycling
  - Worker process health monitoring
- .NET web apps and services are more reliable on IIS6
  - IIS6 offers complete process isolation
  - ASP .NET plugs into new IIS6 features such as REPORT_UNHEALTHY to further improve reliability

Resources

- http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/iis/evaluate/iis6ovw.asp
  - IIS6 Overview

- AWS302 – Internet and Intranet Deployments Scenarios
- AWS306 – IIS Security

Microsoft®

Microsoft Windows.net Server Family